Facial Expression Recognition Via Enhanced Stress Convolution Neural Network for Stress Detection

Wan-Ting Chew, *Siew-Chin Chong, Thian-Song Ong and Lee-Ying Chong

Abstract—**The analysis of facial expressions for human stress detection has recently received wide attention. A facial expression is commonly used to explain a person's emotional state to an observer. Based on a deep learning approach, we propose a lightweight and reliable method to detect stress by using facial expression recognition. This method is called Enhanced Stress Convolutional Neural Network (ESCNN). The ESCNN identifies a person's facial expression and categorizes them into stress or non-stress categories based on their emotional state. Anger, sadness, disgust, and fear are categorized as stress outputs, whereas facial expressions of happiness, neural activity, and surprise are categorized as non-stress outputs. With this approach, Transfer Learning with MobileNet V2 and Tensor Flow is used to run the pre-trained FER2013 dataset and determine stress or no-stress. The MobileNet V2 model will make the proposed method lighter and faster than other algorithms. To get the highest accuracy, different epochs and parameters were tested. After several experiments, ESCNN, the combination of Transfer Learning and Haar Cascade face detection, produced the most effective performance for stress recognition. The experimental analysis vindicated the superiority of ESCNN over state-of-the-art methods.**

Index Terms— **Facial Expression, Convolution Neural Network, Transfer Learning, Stress Detection**

I. Introduction

A person's emotions will fluctuate according to the different situations or be affected by the person they interact with. Technology advancements have introduced facial expression recognition for stress management in psychoanalysis [1] [24]. Facial expressions are characterized by changes in the state of the eyes, face, mouth; for example, the eyes can communicate and express amorous feelings. There are seven common types of facial expression (happiness, sadness, surprise, anger, disgust, fear, and neural), and at least 21 types of human facial expression in the world [2]. Stress negatively affects a person's health, brings negative emotions and, at its worst, may cause mental health problems. Stress statistics collected by Single Care have shown that the most common stresses are similar worldwide, including money, work, and family responsibilities. As of 2020, a new stressor has emerged: the COVID-19 pandemic, a contentious political climate, and more. There are almost 1/3 people around the world who reported feeling stressed, angry, and worried in 2019. In 2017, approximately 284 million people worldwide were affected by anxiety disorder anxiety disorders [3].

Deep learning is a branch of machine learning that uses artificial intelligence (AI) to characterize and learn data. Recently, the deep learning approach has been studied extensively to improve the performance of facial expression recognition systems [4]-[6]. Deep learning supports various methods to recognize facial expressions, such as DeepFace, VGGFace, FaceNet, and the DeepID systems [7]. For machines to analyze facial expressions, Networks (CNN) have been introduced as an approach to teach them to recognize and classify images and videos. There are various of CNN models available in deep learning, including ResNet, VGGNet, GoogleNet, MobileNet, and others. At present, improvement ideas for the CNN model are being developed toward network depth and adding more convolution calculations.

A new method for facial expression recognition was proposed in this work. The proposed method is called Enhanced Stress CNN (ESCNN). The ESCNN runs pretrained models on the FER2013 dataset using MobileNet V2 and Tensor Flow. Based on the analysis of facial expressions, a person is classified as stressed or unstressed. In ESCNN, MobileNet V2 also includes Relu6, Max-pooling layer, Fully Connected Layer, and SoftMax Probabilities. Haar Cascade face detection is also being applied to identify faces in the image. MobileNet V2 is famous for its lightweight model compared with other CNN models such as ResNet50 and Sequential Model [8]. MobileNet V2 reduces 17 MB network and reduces 4.2 million parameters. MobileNet V2 models are smaller and faster than other models with the same accuracy across the entire latency [26]. Benefits of applying MobileNet V2 are:

- 53 layers + 3 layers deep
- lightweight (reduces 17 MB of network size and reduces 4.2 million parameters)
- smaller and faster for the same accuracy across the entire latency

ResNet 50 is more complicated than MobileNet V2 because it has 50 layers deep, and the ResNet 50 model's time

Manuscript received December 30, 2021; revised June 28, 2022.

This work was supported in part by the Internal Research Fund (IR Fund) MMUI/210034, 2021 from Multimedia University.

Wan-Ting Chew is a Research Assistant of Multimedia University, Malaysia (email: wanting0604@gmail.com).

Siew-Chin Chong is a Senior Lecturer of Multimedia University, Malaysia (*corresponding author, phone: +60162221826; fax: +6062318840; e-mail: chong.siew.chin@mmu.edu.my).

Thian-Song Ong is a Professor at Multimedia University, Malaysia (email: tsong@mmu.edu.my).

Lee-Ying Chong is a Lecturer at Multimedia University, Malaysia (email: lychong@mmu.edu.my).

lifetime is uncertain [27]. If we reduce the network to a certain number, there may be a sampling error when completing the training. The highlights of adopting ResNet 50 are listed below:

- 50 layers deep
- increase the performance of neural networks with numerous layers
- uses Batch Normalization as the core to adjust the input layer

Sequential Model is a linear sequential progression, and its structures are easier to create [28]. However, the Sequential Model is not suitable for multiple-input and output. It is also not suitable for layer sharing because it has fewer layers compared to other models. Linear progression through the phases may also result in issues identification being pushed downstream and causing delays [29]. Properties by implementing the Sequential Model are:

- each layer has exactly one input and one output tensor
- not appropriate for multiple input and outputs
- allows building model layer by layer

As a result, the Enhanced Stress CNN (ESCNN) algorithm has been tested on different parameters. This proves that the ESCNN algorithm with MobileNet V2 achieves the best performance than ResNet 50 and Sequential Model. The ESCNN algorithm analyses the facial expressions in the images in seven categories: angry, disgust, fear, happy, neutral, sadness, and surprise [30]. Based on the result of the emotion analysis, the emotion is then categorized into one of two levels, either stress or non-stress. Among the seven categories, anger, disgust, fear, and sadness are grouped as stress results, while happy, neutral, and surprise are grouped as non-stress results [31].

This paper consists of five sections. Section 1 is the introduction of facial expression recognition in detecting emotional stress. Section 2 discusses the literature review of the facial expression methods, Section 3 explains the overall idea of the proposed system, and Section 4 presents the experimental results. In the end, Section 5 is the conclusion and future work of this research. The summary is done to conclude all the steps of this algorithm, pros and cons, comparison, the accuracy of the training epochs, and conclusion.

II. Literature Review

A facial expression is a kind of body language symbol. It expresses an individual's emotional state toward itself or toward the observer [25]. With technological advances, stress detection based on facial expression analysis has become an important research direction for machine learning. Image processing for facial expression recognition is available with two different methods: The Handcrafted (HC) approach and the Deep Learning (DL) approach. The difference between HC approach and DL approaches is defined in Fig. 1.

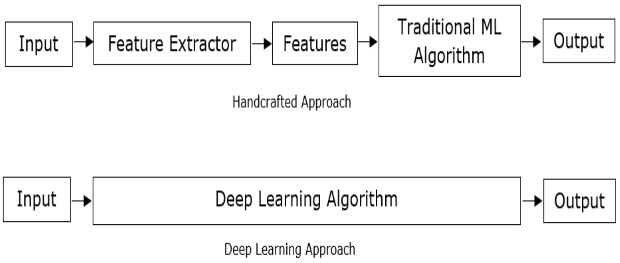


Fig. 1. Different between HC approach and DL approaches

A. *Handcrafted (HC) approach*

The Handcrafted (HC) approach uses a manually predefined algorithm engineered by a data scientist. This enabled the developer to set and tune the features manually [32]. After the configuration model, the developer can control the added features for predicting purposes. For example, using a machine-learning algorithm is to classify the data, perform follow-up analysis, exploring, and finally end with a good performance result [33]. However, the problem with this approach is challenging; it is hard to obtain optimal parameters that can guarantee a good performance. Local Binary Patterns (LBP), Locally Linear Embedding (LLE), and Gabor Wavelet (GW) Transform are examples of this approach. Fig. 2. Below shows the face description with LBP.

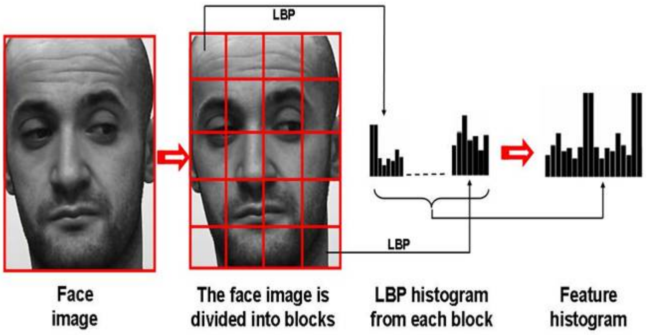

LBP
LBP
Face image
The face image is divided into blocks
LBP histogram from each block
Feature histogram

Fig. 2. Face Description with Local Binary Patterns (LBP)

Local Binary Patterns (LBP) [9] are visual operators used for classification in the field of computer vision. The LBP model is a simple and very effective texture operator. It compares each pixel with its nearby pixels and saves the result as a binary number [34]. At present, LBP has been applied to face recognition, license plate recognition, and other fields. In facial expression recognition, LBP used the feature vector for classification and recognition. If directly extract features from two pictures and perform discriminant analysis, a more significant error will occur due to misaligned positions [35]. Researchers found it possible to divide a picture into several sub-regions, extract the local binary pattern features for each pixel in each sub-region, and then establish local binary pattern features in each sub-region statistical histogram. In this way, each sub-region can be described by a statistical histogram, and the entire picture is composed of several statistical histograms for classification.

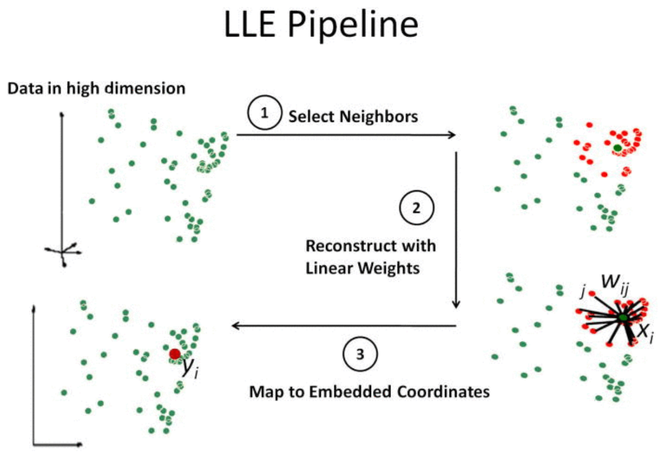


Fig. 3. Step of Locally Linear Embedding (LLE) Mapping Data

Fig. 3. above showing the step of LLE work. Locally Linear Embedding (LLE) [10] is a dimensionality reduction method. LLE is focuses on maintaining the local linear features of the sample during dimensionality reduction. LLE is widely used in image recognition, data visualization, and other fields [36]. The LLE algorithm can be used to reduce the dimensionality of the image and extract the features of the sample set. The main advantage of the LLE algorithm is that it is a locally linear low-dimensional manifold learned from different dimensions [37]. Another advantage of the LLE algorithm is that it only needs one parameter to tune, which is named K. If the K value is too small or too large, the geometry of the original data will be impossible to accommodate.

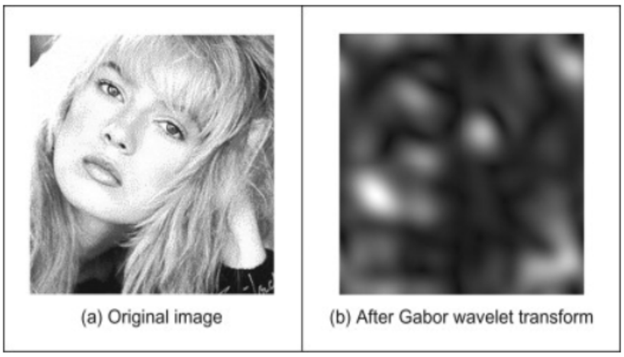


Fig. 4. Example of Gabor Wavelet (GW) transform

Gabor Wavelet (GW) [11] transforms a subset of nodes or all pixels in a face image. It extracts the texture features of the face and performs subsequent feature selection and dimensionality reduction. Due to its similarities to the human visual system, GW is generally considered suitable for facial feature extraction. It has excellent direction selectivity and local structural characteristics from multiple directions in the local area of the image [38]. Simultaneously, it is direction invariance and scale invariance. It uses a joint timescale function to analyze non-stationary signals, effectively extracting information from the signal, and perform multi-resolution refinement of the function or signal through operation functions such as scaling and translation. By choosing the right filter, the GW transform can cover the entire frequency range, reduce or eliminate redundancy between different extracted features. As Fig. 4. Above showing one of the examples how GW transform the image between different extracted features.

B. *Deep Learning (DL) approach*

The Deep Learning (DL) approach is a branch of machine learning. This is an algorithm that uses an artificial neural network as a framework for performing characterization and training [39]. The information obtained during the DL process helps interpret data, such as facial images. To implement DL into facial expression recognition, choose a data-rich expression model and then uses Convolutional Neural Networks (CNN) models, Deep Belief Networks (DBN), Recurrent Neural Networks, and others. DL is a complex machine learning algorithm that achieves good results in image recognition far away from the HC method. There have been several deep learning frameworks being applied in the fields of facial expression and gesture.

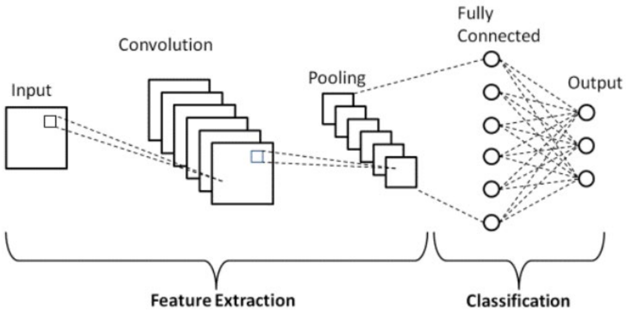


Fig. 5. Basic Architecture of Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) are neural networks that can respond to surrounding units in the coverage area and have excellent performance for large-scale image processing [40]. As shown in Fig. 5. CNN are made up by 3 layers which is convolutional layers, pooling layers, and fully connected (FC) layers. It has a robust classification for image identification and verification. There are some famous CNN models, such as LeNet, AlexNet, VGG, GoogLeNet, and ResNet. The authors [12] proposed using VGG16, VGG19, and ResNet V2 models to run experiments on facial expression recognition systems for stress detection with CNN. This experiment gets the best accuracy for the VGG16 is 0.7665, 0.7257 for the VGG19 model, and 0.8249 for ResNet V2. As a typical network structure in machine learning, CNN has proven to achieve excellent results in image enhancement, image processing, image recognition, and others.

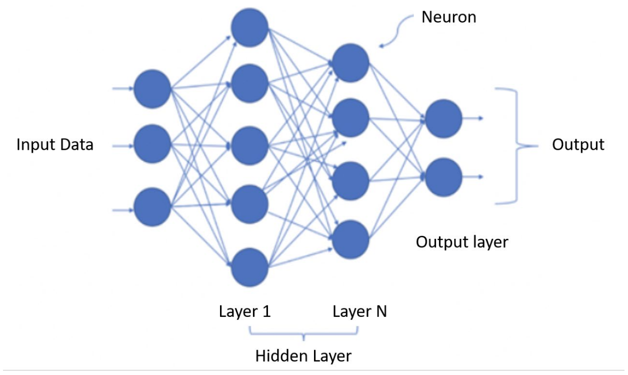


Fig. 6. Structure of Deep Neural Network (DNN)

Fig. 6. Show structure of Deep Neural Network (DNN). DNN is a framework for DL, a neural network with at least one hidden layer. This is similar to Shallow Neural Networks. DNN provides modeling of complex non–linear systems [41]. DNN is a discriminative model that can be trained using backpropagation algorithms. The assumed ratio as a significant component of facial expression can be learned by the trained DNN model. On DNN, the very first neural network architecture for facial expression recognition is EmotionNet Nano. Based on the comparative performance between EmotionNet Nano – A and Nano – B networks in the experiment [13], EmotionNet Nano – A achieve comparable accuracy with the highest – performing state of the art networks more than magnitude larger, while EmotionNet Nano – B had a lower accuracy with the highest-performing networks, but it can still achieve comparable accuracy while being three orders of magnitude smaller concerning to the number of parameters. EmotionNet Nano variants can provide a good combination of accuracy and complexity, suitable for embedded scenarios [42].

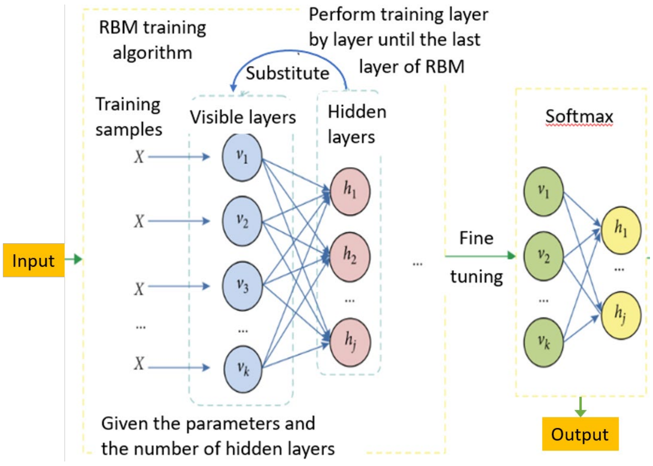


Fig. 7. How Deep Belief Network (DBN) work

The Deep Belief Network (DBN) algorithm is a type of neural network in machine learning which can be used for unsupervised learning and supervised learning [43]. Fig. 7. shows how DBN perform from input image to output. DBN establishes a joint distribution between observed data and labels compared with the traditional neural network of the discriminant model. The entire neural network can generate training data according to the maximum probability, by training the weights between its neurons. For example, an experiment about how DBN performs on facial expression recognition has been performed by [14]. They used LBP combined with improved DBN to run recognition rates in the JAFFE database. In the end, the DBN algorithm achieves the best performance in three datasets compared with other algorithms. Through deep learning networks, the DBN model can extract deeper features and improve recognition rates.

An RNN transfers state cyclically on its network to accept a broader range of time series structure inputs. The advantage of using an RNN is that the temporal dependencies present in the image sequences can be considered during classification. If the entire process is automatic, RNN can be used to make online predictions. The system would be ideal for real-time facial expression recognition [15], and an enhanced version has been done by [16]. However, a Recurrent Neural Network (RNN) is a type of neural network that cannot handle the problem of disappearing gradients with recursion. It is also tricky to capture long-term time correlations. However, this problem can be solved by combining different Long Short-Term Memory (LSTM) types.

III. The Proposed System

Based on the problem defined in Section 1, ESCNN used as a deep evaluation model for stress detection via facial expression. ESCNN produced a fast, lightweight, and high-accurate algorithm in detecting stress with different individual facial expressions. To further improve the accurate of facial expression detection, ESCNN combined Transfer Learning (TL) and MobileNet V2 in Transfer Flow (TF). MobileNet V2 was being applied because it is computationally 35% faster than version 1 with the same accuracy when running object detection and segmentation. As a feature extractor, MobileNet V2 also requires 5.3 times fewer parameters and 5.2 times fewer operations than MobileNet V1 [44]. MobileNet V2 as a pre-trained model offers several benefits, such as its ease of integrating with other TL and TF capabilities. It allows training and experiments to be run faster and it is available for many visual recognition tasks. The TF and TL are construct as in Fig. 8 below.

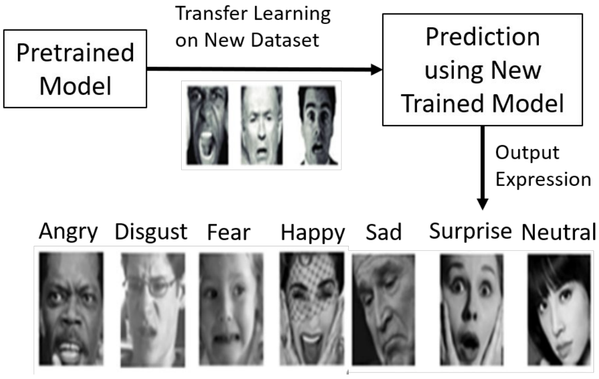


Fig. 8. Transfer Learning using Tensor Flow

The combination of TL and TF also brings many benefits. Before fine-tuning, the TL offers a higher starting point, which allows the models to perform some tasks without training, while TF is scalable, almost any operation can be run on the platform with TF. The combination of TL and TF in ESCNN also improves the computational efficiency of experimental testing. This is because TL is a pre-trained model while TF supports an architecture with a Tensor Processing Unit (TPU), which allows models to be executed faster. Overall, TF and TL help ESCNN to conduct numerous complex features and complete the process faster than previous algorithms. Analysis result of ESCNN will be more accurate and with a better initial model than others. Besides, different algorithms and features also have been added and tested in the ESCNN algorithm. The purpose of adding features is to read a person's emotions more accurately and to achieve the best performance. Overall, the proposed ESCNN algorithm has been done as in Fig. 9 below.

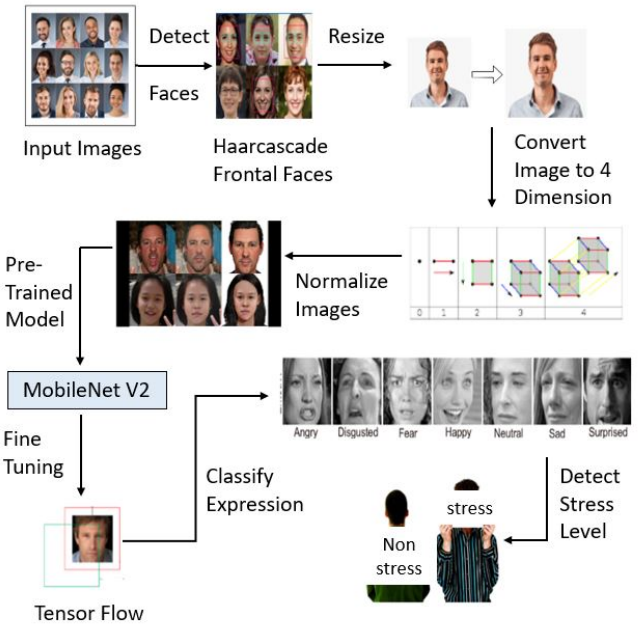


Fig. 9. Overview of the proposed ESCNN algorithm.

First, input a facial image in Haar Cascade Frontal Face detection. Haar Cascade Frontal will crop a rectangular shape if they detect face in the image [45]. If no facial is being detected, the algorithm will show an "Error". Image in Haar Cascade will be converted to gray image because Haar Cascade only works on gray images, not in RGB images. After finding all possible faces inside the images, the image will be plotted and cropped until the left faces in the picture.

Second, change the size of the image to (224, 224, 3). The purposes of changing the image size is because Deep Learning (DL) classifier in Transfer Learning (TL) requests all the file size size to be 224 x 224 pixels. In ESCNN, MobileNet V2, one of the DL classifiers in TL, also accepts an image size of 224 x 224. After converting to 224 x 224 pixels, then it will convert the image again into four dimensions. This is because MobileNet V2 uses four dimensions in depth-wise convolutional neural networks. Next, normalize the images by dividing them by 255 square pixels. By printing the image array, it will show a series of numbers. The larger number for the series number is 255 square pixels, and the smaller number is 0, due to the maximum level for the black being 255, and the minimum level for the white is 0. Thus, this proposed algorithm is divided into 255 square pixels, so that the black is normalized. The TL formula is defined by Eq. (1) Below:

$$T = \{Y, f(x)\}, \mathrm{Ds} \neq \mathrm{Dt}, \text{ or } \mathrm{Ts} \neq \mathrm{Tt} \qquad (1)$$

f is used to predict the $f(x)$ of a new instance of x, while the label is a label space. Ds are a source domain, and Ts is a learning task. A target domain, Dt, and a learning task, Tt are not equal [46].

Then, image will be transfer to MobileNet V2 model. In MobileNet V2 model, image will run from the base input with 0, which means the first layer of the MobileNet V2 [47]. In Python, the last layer means "-1". At ESCNN, it is only necessary to take seven classes. Since the last layer is only suitable for thousands of classes, the last layer is being cut down by base output [-2]. After cutting it down, the last layer should be the (global_average_pooling2d) layer. The MobileNet V2 structure is defined in Fig. 10.

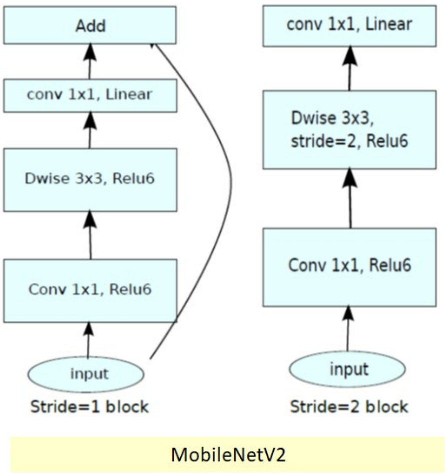


Fig. 10. Structure of the MobileNet V2.

The output of the image is passed from the (global_average_pooling2d) layer into the final layer. Three new layers have been created and added to this algorithm as the last output.

$$\text{tf.layers.dense(args)} \qquad (2)$$

Dense layer is an inbuilt function of Tensorflow.js library. It is used to create fully connected layers, and (args) is used to pass a variable number of arguments to a function. Formula for Dense layer is shown as described above in Eq. (2)

The first two layers are layers.dense 128 and layer.dense 64 applied with Relu6 activation. Classes of ESCNN are only 7, therefore the last layer, layer.dense 7, with using SoftMax activation. This is because SoftMax is a classification layer, it is the last fully connected layer to be activated [48]. The SoftMax formula is defined by Eq. (3) Below:

$$\sigma\left(\vec{z}\right)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (3)$$

σ is meant as SoftMax, while Z is the input vector for SoftMax. The standard exponential function of the input vector divides the sum of the K number of classes in the multi-class classifier. It multiplies with the standard exponential function of (Z_i) for the output vector. SoftMax turns arbitrary real values into probabilities and uses each number exponentially as a numerator [49].

In the end, because the image is in number rather than one hot encoding, we chose "sparse categorical cross entropy" as the loss and "adam" as the optimizer. The accuracy of the images starts at 40% because it is a TL model, and it tunes the weights accordingly. As a result, ESCNN generates output in seven different categories. By using the results of facial emotion, then the system will further categories it as stress or non-stress output. Happy, Neutral, and Surprise are labeled as Non-Stress output, while Sad, Angry, Disgust and Fear are labeled as Stress output.

IV. Experimental Results and Discussion

FER 2013 dataset as one of the famous datasets in the world was being applied in ESCNN. It is being used to train and collect facial emotions in this research [17]. There are 958 samples of angry faces, 111 samples of disgust faces, 1024 samples of fearful face, 1774 samples of happy faces, 1233 samples of neutral faces, 1247 samples of sad faces, and 831 samples of surprise faces in the test directory. The training directory contains 3995 samples of angry faces, 436 samples of disgust faces, 4097 samples of fearful faces, 7215 samples of happy faces, 4965 samples of neutral faces, 4830 samples of sad faces, and 3171 samples of surprise faces [50]. Examples of the dataset FER2013 are shown in Fig. 11 below. Various facial expressions were included in the data.

Fig. 11. Sample images of FER2013 Dataset.

After the FER2013 dataset's images are read and input into our proposed method ESCNN, the Haar Cascade face detection algorithm is used to detect each facial expression in the images. The major task of face detection is to identify the faces in the face images [18], [19]. Haar Cascade uses edge, line, and four rectangle detection features to calculate pixel intensities and calculate the differences. Integrate images are created to speed up the calculation of the Haar Cascade features. Sub-rectangles and array references are also created to compete Haar Cascade features. To choose the best features and train the classifiers, Adaboost training creates a robust classifier and a weak learner to detect the objects. In the final stage, the weak learner trains the image by using boosting to get a highly accurate classifier. Then, it will identify whether the object is positive or negative. The flowchart of Cascade Classifiers is shown in Fig. 12.

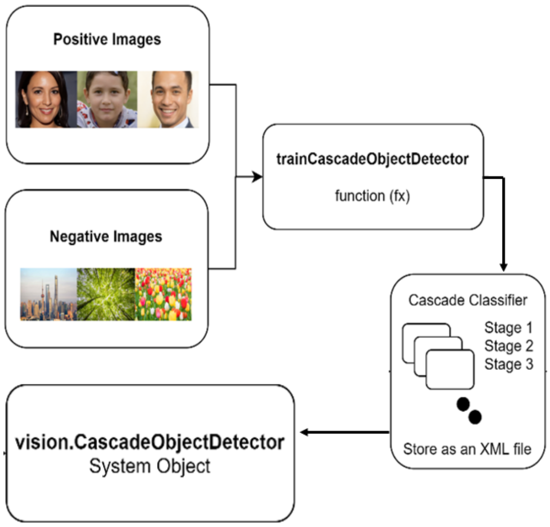
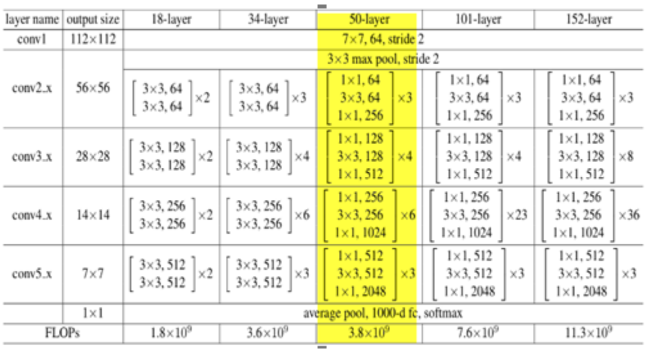


Fig. 12. Process flow of Cascade classifiers.

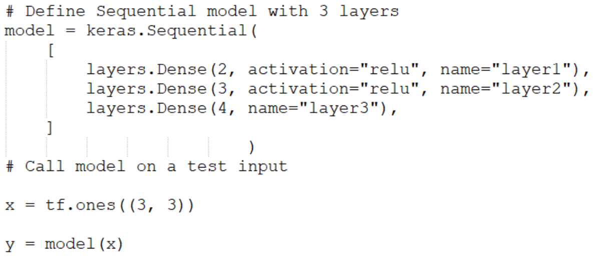
In ESCNN, different models and numbers of epochs have been run and tested to obtain the best performance. The experiments can be categorized into two types. First will calculate facial expression accuracy, while the second will calculate stress detection accuracy. For the first experiment, a performance comparison was made between the ESCNN algorithm, the ResNet 50 model, and the Sequential Model. ResNet 50 and Sequential Model are being compared with ESCNN because ResNet is one of the most famous models in deep learning. Fig. 13. Below shows structure for the ResNet 50



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2.x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3.x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4.x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5.x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 13. ResNet 50 Structure.

ResNet enables the model to be trained from a hundred to thousands of layers, and it excels at object detection and face recognition. ResNet have 5 type of model which is 18-layer , 34-layer , 50-layer, 101-layer and 152-layer. In this experiment, ResNet 50-layer were being select as one of the comparison models. It contains convolution with a kernel size of 7 x 7 and 64 different kernels all with a stride of size 2 giving us 1 layer. In the next convolution there is a 1 x 1,64 kernel following this a 3 x 3,64 kernel and at last a 1 x 1,256 kernel. These three layers are repeated in total 3 time so giving us 9 layers in this step. Next, we see kernel of 1 x 1,128 after that a kernel of 3 x 3,128 and at last a kernel of 1 x 1,512 this step was repeated 4 time so giving us 12 layers in this step. After that, there is a kernel of 1 x 1,256 and two more kernels with 3 x 3,256 and 1 x 1,1024 and this is repeated 6 time giving us a total of 18 layers. And then again, a 1 x 1,512 kernel with two more of 3 x 3,512 and 1 x 1,2048 and this was repeated 3 times giving us a total of 9 layers. After that, we do an average pool and end it with a fully connected layer containing 1000 nodes and at the end a SoftMax function, so this gives us 1 layer. So, the totaling is 1 + 9 + 12 + 18 + 9 + 1 = 50 layers Deep Convolutional network



```
# Define Sequential model with 3 layers
model = keras.Sequential(
    [
        layers.Dense(2, activation="relu", name="layer1"),
        layers.Dense(3, activation="relu", name="layer2"),
        layers.Dense(4, name="layer3"),
    ]
                                  )
# Call model on a test input

x = tf.ones((3, 3))

y = model(x)
```

Fig. 14. Sequential Model Structure

Fig. 14. Above showing how was the Sequential Model Structure. A Sequential model is appropriate for a plain stack

of layers where each layer has exactly one input tensor and one output tensor. It can train the model in the fastest way possible. Since the Sequential Model is a linear progression model, the structure model is easier to construct and run than other models [51]. This experiment applied the FER 2013 dataset to each model to train the model in 1 epoch, 5 epochs, 10 epochs, and 15 epochs, respectively. One epoch is the learning cycle. The number of epochs defines the number of learning algorithms will work through the entire dataset. The higher the epoch, the higher the accuracy.

Among the performance of three different models which is shown in Fig. 15, the ESCNN algorithm outperforms ResNet 50 and Sequential Model for epoch 1, with an accuracy of 0.5323 (53.23%), while ResNet 50 and Sequential Model achieve 0.4659 (46.59%) and 0.3205 (32.05%) respectively.

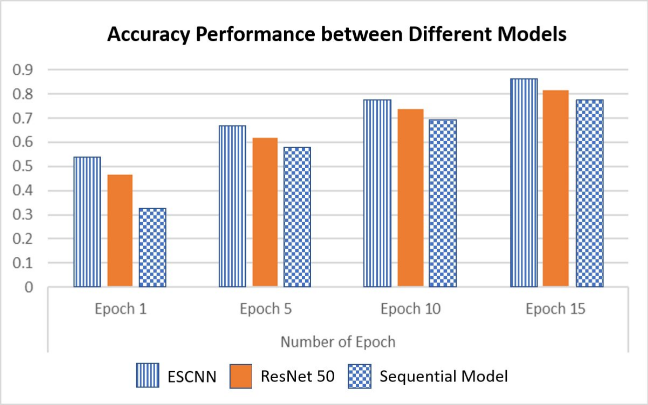


Fig. 15. Accuracy Performance between 3 different models.

In the experiment run over 10 epochs, the ESCNN algorithm again achieves better accuracy than ResNet 50 and the Sequential Model. ESCNN got 0.8554 (85.54%) accuracy, while ResNet 50 got 0.8167 (81.67%), while the low accuracy is 0.7732 (73.32%) from the Sequential Model. Even though Sequential has made a huge improvement compared with epoch 5, it still generates bad accuracy compared with ESCNN and ResNet 50. In an experiment running for 15 epochs, the ESCNN algorithm once again achieves a better result than the other models. ESCNN achieves 0.773 (77.3%) accuracy, while ResNet 50 and Sequential Model achieve 0.7373 (73.73%) and 0.6931 (69.31%) respectively.

After running with epoch 15, ESCNN achieves better accuracy results, then ResNet 50 and the Sequential Model. The best ESCNN result that can be obtained in epoch 15 is (0.773) 77.3%. The Sequential Model has the worst result because only a few layers are created in this model. ESCNN manages to score higher accuracy results because three new layers are added in MobileNet V2 of the ESCNN algorithm. The original version of MobileNet V2 contains 53 layers, while ResNet 50 contained 50 layers. After adding 3 new layers to the original MobileNet V2 used by ESCNN, there are 56 layers. This allows the process to extract more features and improve the accuracy.

After running the first experiment, the second experiment was followed up to test the accuracy of stress detection of ESCNN based on a facial expression recognized in the first experiment. ESCNN analyses the facial expression and categorizes the input faces as Stress or Non-Stress. The purpose of running the second experiment was to observe the accuracy of the ESCNN in classifying Stress and Non-Stress based on facial expressions. Before running this experiment, the dataset from FER2013 was reconstructed from 7 categories to 2 categories: Stress files, and Non-Stress files. "Happy, Neutral, and Surprise" face images are categorized as a Non - Stress class, while "Sad, Angry, Disgust, and Fear" images are categorized as Stress class. Fig. 16 shows the pipeline of facial expressions to stress recognition.

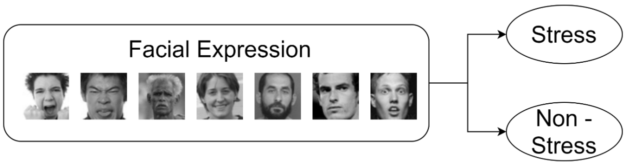


Fig. 16. Facial Expression to Stress Recognition.

As referred to in Table I, the second experiment is named as Stress Recognition Accuracy, while the first experiment is named as Facial Expression Accuracy. The same experiment was carried out for Stress Recognition Accuracy and it was discovered that Stress Recognition Accuracy achieves a better result than Facial Expression Accuracy. The objective of the second experiment was to test the proposed ESCNN in classifying the various facial expression images to the correct stress categories. Here, we name the two categories Stress and Non-Stress. In epoch 1, Stress Recognition Accuracy gets a higher result than Facial Expression Accuracy, 0.7506 (75.06%) and 0.5323 (53.23%). In epoch 5, Stress Recognition Accuracy again achieves a better result than Facial Expression Accuracy, 0.8383 (83.83%) and 0.6677 (66.77%). The best performance for ESCNN is in epoch 15. It achieves up to 0.95 (95%) stress accuracy, which is the highest accuracy run in this whole experiment. This proves that ESCNN achieves the promised results as proposed before.

TABLE I
COMPARISON OF ESCNN ON FACIAL EXPRESSION ACCURACY AND STRESS ACCURACY

ESCNN	Epoch 1	Epoch 5	Epoch 10	Epoch 15
Facial Expression Accuracy	0.5323	0.6677	0.773	0.8554
Stress Recognition Accuracy	0.7506	0.8383	0.9047	0.95

From the experiment, algorithm performance can be affected by the dataset. To avoid mistakenly changing the original dataset, the algorithm will usually make a replicate of the dataset that is used to train. Referring to Table I, the accuracy of Facial Expression is lower than Stress Recognition Accuracy. This is because Stress Recognition Accuracy reconstructs the FER2013 dataset from 7 categories to 2 categories. Therefore, the Stress Recognition Accuracy of ESCNN reduces the numbers of copying dataset files from 7 categories to 2 categories. This helps ESCNN train models more accurately and faster. Thus, the Stress Accuracy of ESCNN performs better compared to the ResNet 50, Sequential Model and Facial Expression Accuracy of ESCNN. Overall, the Stress Recognition of ESCNN achieves the highest accuracy, which is 0.95 (95%) accuracy after running 15 epochs. The result is still considered satisfactory.

Fig. 7 shows the Receiver Operating Characteristic curve (ROC) obtained from ESCNN with the FER2013 dataset. This shows the performance of ESCNN at all classification thresholds. The ROC curve includes a True Positive Rate (TPR) and a False Positive Rate (FPR). Below is the formula for TPR and FPR is expressed in Eq. (4):

$$TPR = \frac{TP}{(TP+FN)}, FPR = \frac{FP}{(FP+TN)} \quad (4)$$

The ROC curve starts from the lower-left corner at sensitivity 0 with the corresponding cutoff to 1 and ends in the upper right with sensitivity 1 with the corresponding cutoff to 0. The closer the ROC curve gets to the upper left corner, the better. In Fig. 17, the ROC curve is closer to the upper right corner, which means that the curve has higher specificity with higher sensitivities. The blue “---” line is the result of assigning cases to the default vs nondefault group randomly.

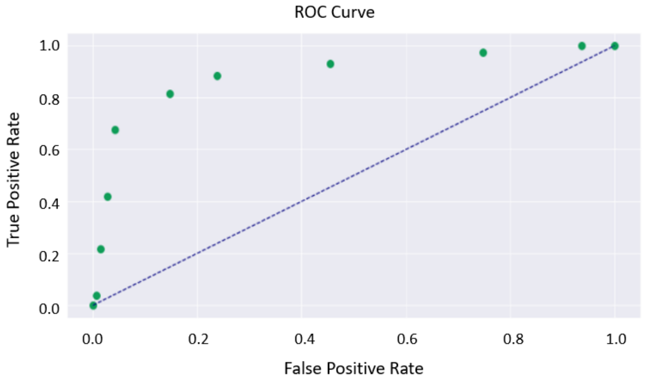


Fig. 17. ROC Curve of ESCNN.

Furthermore, a performance comparison of the ESCNN in facial expression tasks with the state-of-the-art methods is observed in Table II. This table shows a performance comparison using the FER2013 dataset, in which the proposed ESCNN outperforms other state-of-the-art methods. To be fair in comparison, only those state-of-the-art approaches that are implemented using FER2013 are listed in Table II for performance comparison.

TABLE II
PERFORMANCE COMPARISON WITH THE STATE-OF-ART-METHODS USING FER2013 DATASET

Method	Accuracy Rate
GoogleNet [20]	65.20 %
VGG + SVM [21]	66.31%
Attentional ConvNet [22]	70.02%
VGG [23]	73.28 %
ESCNN (proposed work)	85.84 %

It is observed that GoogleNet [20] and VGG + SVM [21] have poorer accuracy rates of 65.20% and 66.31%, respectively. Simultaneously, Attentional ConvNet [22] and VGG [23] appear to generate slightly better results with the accuracy of 70.02% and 73.28%, respectively. In contrast, ESCNN achieves the best performance among all listed methods, which is 85.84 %. Compared with others, ESCNN used MobileNet V2 to train the model with 56 structure layers, while GoogleNet has only 22 layers and VGG has 19 layers [52]. Along with Tensor Flow, it helps in data visualization, reduces the effort to visit all the codes, and effectively resolves the neural network. Furthermore, it allows the building of neural network graphs representing operations as nodes, which brings great impact on the overall design of ESCNN in analyzing facial expressions.

V. CONCLUSION

This paper proposed a stress detection method through facial expressions, namely, ESCNN. In ESCNN, different kind of algorithm and features being combined together as a new way to propose a brand-new algorithm. Haar Cascade Frontal Face is one of the features that being added in ESCNN for the purpose detecting face. MobileNet V2 is also being used as a pre-trained model for feature extraction, and Tensor Flow is used for fine-tuning the model. For the performance evaluation, the FER2013 dataset was used to execute the experiments with varied settings. Seven facial expression types are included in the FER2013 dataset (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral). Two main experiments have been conducted in this paper: the first experiment was to analyze the accuracy of Facial Expression, and the second experiment was to estimate stress. Compared to ResNet 50 and Sequential Model, ESCNN proves its efficiency and stability by running faster in the experiments. The size of the ESCNN is also smaller than ResNet 50 and the Sequential Model. This is because ESCNN used the MobileNet V2 model, which is a well-known lightweight model. This can reduce the network size by 17 MB and its parameters by 4.2 million [53]. In ESCNN, Relu6 and SoftMax are also being applied for layer activation in the MobileNet V2 model. For the final classification layer, SoftMax is applied. ESCNN has proven that it produces the best performance compared with ResNet 50 and Sequential Model. ESCNN achieves the best result, which is 95% for Stress Recognition Accuracy (15 epoch) and 85.54% accuracy of Facial Expression (15 epoch), as compared with ResNet 50 (81.67%) and Sequential Model (77.32%). Overall, ESCNN is viable for stress detection and recognition of facial expressions. This result confirms that ESCNN performance is better than other state-of-the-art methods.

REFERENCES

[1] D. Venkataraman, Namboodiri Sandhya Parameswaran, “Extraction of Facial Features for Depression Detection among Students,” in Amrita School of Engineering, vol. 118. no. 7, pp. 455-463, 2018.
[2] Shichuan Du, Yong Tao, and Aleix Martinez, “Compound facial expressions of emotion,” Proceedings of the National Academy of Sciences of the United States of America, vol. 111, no. 15, 2014.
[3] SingleCare Team, “Mental health statistics 2021,” the CheckUp, 2021.
[4] C. Liao, R. Chen and S. Tai, "Emotion stress detection using EEG signal and deep learning technologies," 2018 IEEE International Conference on Applied System Invention (ICASI), pp. 90-93, 2018.
[5] M. I. Georgescu, R. T. Ionescu, and M. Popescu, “Local learning with deep and handcrafted features for facial expression recognition,” IEEE Access, vol. 7, 2019.

[6] M. K. Pramerdorfer, Christopher, "Facial expression recognition using convolutional neural networks: state of the art," arXiv, vol. 1612.02903, 2016.

[7] Jason Brownlee, "A Gentle Introduction to Deep Learning for Face Recognition," Deep Learning for Computer Vision, 2019.

[8] F. "The differences between Inception, ResNet, and MobileNet. Medium," https://medium.com/@fransiska26/the-differences-between-inception-resnet-and-mobilenet-e97736a709b0, 2019.

[9] Juefei-Xu, Felix, and Marios Savvides. "Learning to Invert Local Binary Patterns." In BMVC. 2016.

[10] Wang J. "Locally Linear Embedding. In: Geometric Structure of High-Dimensional Data and Dimensionality Reduction," Springer, Berlin, Heidelberg, pp. 203-220, 2012.

[11] Qian Zhang, Haigang Li, Ming Li, Lei Ding. Feature extraction of face image based on LBP and 2-D Gabor wavelet transform[J]. Mathematical Biosciences and Engineering, vol. 17, no. 2, pp. 1578-1592, 2020.

[12] Jose Almeida and Fatima Rodrigues, "Facial Expression Recognition System for Stress Detection with Deep Learning," Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021), vol. 1, pp. 256-263, 2021.

[13] Lee JR, Wang L and Wong A (2021) EmotionNet Nano: An Efficient Deep Convolutional Neural Network Design for Real-Time Facial Expression Recognition. Front. Artif. Intell. 3:609673.

[14] Wu, Yao & Qiu, Weigen, "Facial expression recognition based on improved deep belief networks," AIP Conference Proceedings, vol. 1864, 2017.

[15] Graves, Alex, Christoph Mayer, Matthias Wimmer, Jürgen Schmidhuber, and Bernd Radig, "Facial expression recognition with recurrent neural networks," In Proceedings of the International Workshop on Cognition for Technical Systems, 2008.

[16] Amr Badr, Mahmoud Khalil and Hazem Abbas "Emotion Recognition by Facial Features using Recurrent Neural Networks," Proceedings of 13th International Conference on Computer Engineering and Systems, pp. 417-422, 2018.

[17] Carrier, P. L., Courville, A., Goodfellow, I. J., Mirza, M., and Bengio, Y, "FER-2013 face database," University de Montreal, 2013.

[18] Vestiana Aza, Indrabayu, Intan Sari Areni, "Face recognition using local binary pattern histogram for visually impaired people," International Seminar on Application for Technology of Information and Communication (iSemantic), 2019.

[19] Indrasom Gangopadhyay, Anulekha Chatterjee, Indrajit Das, "Face detection and recognition using Haar classifier and LBP histogram," Int. J. Adv. Res. Comput. Sci., vol. 9, no. 2, 2018.

[20] P. Giannopoulos, I. Perikos, and I. Hatzilygeroudis, "Deep learning approaches for facial emotion recognition: A case study on FER-2013," in Smart Innovation, Systems and Technologies, vol. 85, 2018.

[21] M. I. Georgescu, R. T. Ionescu, and M. Popescu, "Local learning with deep and handcrafted features for facial expression recognition," IEEE Access, vol. 7, 2019.

[22] S. Minaee and A. Abdolrashidi, "Deep-emotion: facial expression recognition using the attentional convolutional network," arXiv. 2019, DOI: 10.3390/s21093046.

[23] M. K. Pramerdorfer, Christopher, "Facial expression recognition using convolutional neural networks: state of the art," arXiv, vol. 1612.02903, 2016.

[24] Santamaria-Granados, L., Munoz-Organero, M., Ramirez-Gonzalez, G., Abdulhay, E., and Arunkumar, N. J. I. A. (2018). Using deep convolutional neural network for emotion detection on a physiological signals dataset (AMIGOS). IEEE Access 7, 57–67. doi: 10.1109/ACCESS.2018.2883213

[25] Maydych, V., Claus, M., Watzl, C., and Kleinsorge, T. (2018). Attention to emotional information is associated with cytokine responses to psychological stress. Front. Neurosci. 12:687.

[26] Sandler, M. (2018, 13 januari). MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv.Org. Geraadpleegd op 27 maart 2022, van https://arxiv.org/abs/1801.04381

[27] Boesch, G. (2022, 17 januari). Deep Residual Networks (ResNet, ResNet50) – Guide in 2022. Viso.Ai. Geraadpleegd op 27 maart 2022, van https://viso.ai/deep-learning/resnet-residual-neural-network/

[28] Team, K. (2020). Keras documentation: The Sequential model. The Sequential Model. Geraadpleegd op 27 maart 2022, van https://keras.io/guides/sequential_model/#:%7E:text=A%20Sequential%20model%20is%20appropriate,%22layer1%22)%2C%20layers.

[29] Lendave, V. (2021, 17 november). A Tutorial on Sequential Machine Learning. Analytics India Magazine. Geraadpleegd op 27 maart 2022, van https://analyticsindiamag.com/a-tutorial-on-sequential-machine-learning/

[30] Xu, Q. (2017). Facial Expressions in Context: Electrophysiological Correlates of the Emotional Congruency of Facial Expressions and Background Scenes. Frontiers. Geraadpleegd op 27 maart 2022, van https://www.frontiersin.org/articles/10.3389/fpsyg.2017.02175/full

[31] Detecting Negative Emotional Stress Based on Facial Expression in Real Time. (2019, 1 juli). IEEE Conference Publication | IEEE Xplore. Geraadpleegd op 27 maart 2022, van https://ieeexplore.ieee.org/document/8868735

[32] Handcrafted vs. non-handcrafted features for computer vision classification. (2017, 1 november). ScienceDirect. Geraadpleegd op 27 maart 2022, van https://www.sciencedirect.com/science/article/abs/pii/S0031320317302224

[33] Wahid, M. F. (2022, 25 januari). A Classical Approach to Handcrafted Feature Extraction Techniques. . . arXiv.Org. Geraadpleegd op 27 maart 2022, van https://arxiv.org/abs/2201.10102

[34] Research and Perspective on Local Binary Pattern. (2013, 1 juni). ScienceDirect. Geraadpleegd op 27 maart 2022, van https://www.sciencedirect.com/science/article/abs/pii/S1874102913600518

[35] A survey on facial recognition based on local directional and local binary patterns. (2018, 1 maart). IEEE Conference Publication | IEEE Xplore. Geraadpleegd op 27 maart 2022, van https://ieeexplore.ieee.org/document/8368757

[36] M. (2021, 12 december). Locally Linear Embedding (LLE) | Data Mining and Machine Learning. Medium. Geraadpleegd op 27 maart 2022, van https://medium.com/analytics-vidhya/locally-linear-embedding-lle-data-mining-b956616d24e9

[37] Dobilas, S. (2022, 5 februari). LLE: Locally Linear Embedding — A Nifty Way to Reduce Dimensionality in Python. Medium. Geraadpleegd op 27 maart 2022, van https://towardsdatascience.com/lle-locally-linear-embedding-a-nifty-way-to-reduce-dimensionality-in-python-ab5c38336107

[38] Face Recognition Using Gabor Wavelet Features with PCA and KPCA - A Comparative Study. (2015, 1 januari). ScienceDirect. Geraadpleegd op 27 maart 2022, van https://www.sciencedirect.com/science/article/pii/S1877050915019638

[39] Lin, S. (2021, 14 december). Deep Learning Research and How to Get Immersed - Towards Data Science. Medium. Geraadpleegd op 27 maart 2022, van https://towardsdatascience.com/deep-learning-research-and-how-to-get-immersed-8bab98c20577

[40] Understanding of a convolutional neural network. (2017, 1 augustus). IEEE Conference Publication | IEEE Xplore. Geraadpleegd op 27 maart 2022, van https://ieeexplore.ieee.org/document/8308186

[41] Burns, E., & Brush, K. (2021, 29 maart). deep learning. SearchEnterpriseAI. Geraadpleegd op 27 maart 2022, van https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network

[42] EmotionNet Nano: An Efficient Deep Convolutional Neural Network. (2021, 21 januari). Vision and Image Processing Lab. Geraadpleegd op 27 maart 2022, van https://uwaterloo.ca/vision-image-processing-lab/publications/emotionnet-nano-efficient-deep-convolutional-neural-network

[43] O'Connor, P. (2013). Real-time classification and sensor fusion with a spiking deep belief network. Frontiers. Geraadpleegd op 27 maart 2022, van https://www.frontiersin.org/articles/10.3389/fnins.2013.00178/full

[44] MobileNetV2: The Next Generation of On-Device Computer Vision Networks. (2018, 3 april). Google AI Blog. Geraadpleegd op 27 maart 2022, van https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html#:%7E:text=Overall%2C%20the%20MobileNetV2%20models%20are,all%20while%20achieving%20higher%20accuracy.

[45] Facial recognition using Haar cascade and LBP classifiers. (2021, 1 november). ScienceDirect. Geraadpleegd op 27 maart 2022, van https://www.sciencedirect.com/science/article/pii/S2666285X21000728

[46] Brownlee, J. (2019, 16 september). A Gentle Introduction to Transfer Learning for Deep Learning. Machine Learning Mastery. Geraadpleegd op 27 maart 2022, van https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[47] Papers with Code - MobileNetV2 Explained. (2019). MobilenetV2. Geraadpleegd op 27 maart 2022, van https://paperswithcode.com/method/mobilenetv2

[48] tf.keras.layers.Dense | TensorFlow Core v2.8.0. (2020). TensorFlow. Geraadpleegd op 27 maart 2022, van https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense

[49] Papers with Code - Softmax Explained. (2019). Softmax. Geraadpleegd op 27 maart 2022, van https://paperswithcode.com/method/softmax

[50] The Facial Emotion Recognition (FER-2013) Dataset for Prediction System of Micro-Expressions Face Using the Convolutional Neural Network (CNN) Algorithm based Raspberry Pi. (2020, 3 november). IEEE Conference Publication | IEEE Xplore. Geraadpleegd op 27 maart 2022, van https://ieeexplore.ieee.org/abstract/document/9288560

[51] Hollemans, M. (2018). MobileNet version 2. Mobilnet V2 Version. Geraadpleegd op 27 maart 2022, van https://machinethink.net/blog/mobilenet-v2/

[52] Sharma, P. (2020, 20 april). Purpose of different layers in a Deep Learning Model. OpenGenus IQ: Computing Expertise & Legacy. Geraadpleegd op 27 maart 2022, van https://iq.opengenus.org/purpose-of-different-layers-in-ml/

[53] Pröve, P. (2018, 5 juli). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Medium. Geraadpleegd op 27 maart 2022, van https://towardsdatascience.com/mobilenetv2-inverted-residuals-and-linear-bottlenecks-8a4362f4ffd5

Wan-Ting Chew received her B.I.T. (Hons) (Security Tech.) from Multimedia University Malaysia in 2021. Currently, she is a Research Assistance in a funded project. Her research interest falls within the biometrics security domain and she had completed a mobile application utilizing an emotion detection engine through facial expression and copyright has been affirmed for this.

Siew-Chin Chong (IEEE Senior Member) received her B.IT (Software Engineering), M. Sc (Information Technology), and Ph.D. (I.T) from Multimedia University in the Year 2003, 2006, and 2018 respectively. She is currently serving as the Deputy Dean of Student Experience & Alumni in the Faculty of Information Science and Technology, Multimedia University Malaysia. Her research interests include machine learning, biometrics security, and mobile app development. She has contributed several indexed journal and conference publications in these domains. She has served as an Editorial Board Member of Progress in Human-Computer Interaction 2018 – 2020 and a committee member of several international conferences.

Thian-Song Ong is a Professor of the Faculty of Information Science and Technology at Multimedia University, Malaysia. His research interests include Biometrics and Information Security. He received an MSc degree in 2001 from the University of Sunderland, the UK, and a Ph.D. degree in 2008 from Multimedia University, Malaysia. He has published more than 60 international refereed journals and conference articles in the related fields. He was on the editorial board for IEEE Biometrics Council Newsletter from 2013-2015. Currently, he has served as the permanent member of the technical committee in SIRIM Malaysia for biometrics standardization in Malaysia.

Chong Lee Ying received her Ph.D. from Multimedia University, Malaysia, in 2018. She is a lecturer with the Faculty of Information Science and Technology, Multimedia University, since 2008. Her current research interests include image processing, machine learning, and biometric authentication. She is also served as an IEEE Senior Member.